|  |  |
| --- | --- |
| Name: Kunaal Umrigar | Roll no: I059 |
| Course: MBA TECH IT | Batch: A3 |
| Date of submission:17th april’18 | Grade: |

**MPMC PROJECT**

org 100h

.model small

.stack 100h

.data

number db 169d

CR equ 13d

LF equ 10d

prompt db CR, LF,'Please enter a valid number : $'

lessMsg db CR, LF,'Value if Less ','$'

moreMsg db CR, LF,'Value is More ', '$'

equalMsg db CR, LF,'You have made fine Guess!', '$'

overflowMsg db CR, LF,'Error - Number out of range!', '$'

retry db CR, LF,'Retry [y/n] ? ' ,'$'

guess db 0d

errorChk db 0d

param label Byte

.code

start:

MOV ax, 0h

MOV bx, 0h

MOV cx, 0h

MOV dx, 0h

MOV BX, OFFSET guess

MOV BYTE PTR [BX], 0d

MOV BX, OFFSET errorChk

MOV BYTE PTR [BX], 0d

MOV ax, @data

MOV ds, ax

MOV dx, offset prompt

MOV ah, 9h

INT 21h

MOV cl, 0h

MOV dx, 0h

while:

CMP cl, 5d

JG endwhile

MOV ah, 1h

INT 21h

CMP al, 0Dh

JE endwhile

SUB al, 30h

MOV dl, al

PUSH dx

INC cl

JMP while

endwhile:

DEC cl

CMP cl, 02h

JG overflow

MOV BX, OFFSET errorChk

MOV BYTE PTR [BX], cl

MOV cl, 0h

while2:

CMP cl,errorChk

JG endwhile2

POP dx

MOV ch, 0h

MOV al, 1d

MOV dh, 10d

while3:

CMP ch, cl

JGE endwhile3

MUL dh

INC ch

JMP while3

endwhile3:

MUL dl

JO overflow

MOV dl, al

ADD dl, guess

JC overflow

MOV BX, OFFSET guess

MOV BYTE PTR [BX], dl

INC cl

JMP while2

endwhile2:

MOV ax, @data

MOV ds, ax

MOV dl, number ; load original 'number' to DL

MOV dh, guess ; load guessed 'number' to DH

CMP dh, dl ; compare DH and DL

JC lower

JE equal

JG greater

equal:

MOV dx, offset equalMsg

MOV ah, 9h

INT 21h ; DOS INT 21h (DOS interrupt)

JMP exit ; JUMP to end of the program

greater:

MOV dx, offset moreMsg ; load address of 'moreMsg' message to DX

MOV ah, 9h

INT 21h

JMP start

lower:

MOV dx, offset lessMsg

MOV ah, 9h

INT 21h

JMP start

overflow:

MOV dx, offset overflowMsg

MOV ah, 9h

INT 21h

JMP start

exit:

retry\_while:

MOV dx, offset retry

MOV ah, 9h

INT 21h

MOV ah, 1h

INT 21h

CMP al, 6Eh

JE return\_to\_DOS

CMP al, 79h

JE restart

JMP retry\_while

retry\_endwhile:

restart:

JMP start

return\_to\_DOS:

MOV ax, 4c00h

INT 21h

end start

RET

ret

